



①9 BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENTAMT

⑫ **Offenlegungsschrift**  
⑩ **DE 43 21 631 A 1**

⑤1 Int. Cl.<sup>6</sup>:  
**G 05 B 19/18**

②1 Aktenzeichen: P 43 21 631.5  
②2 Anmeldetag: 30. 6. 93  
④3 Offenlegungstag: 12. 1. 95

⑦1 Anmelder:  
Robert Bosch GmbH, 70469 Stuttgart, DE

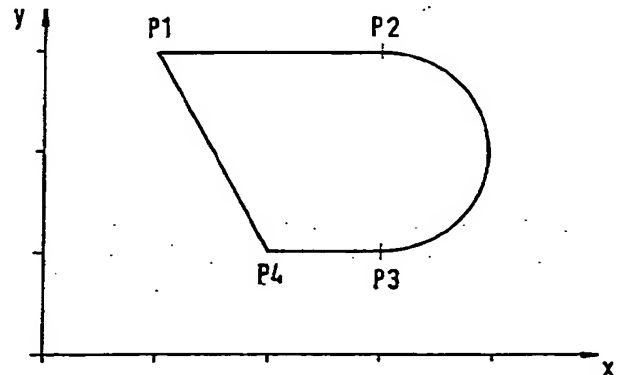
⑦2 Erfinder:  
Erfinder wird später genannt werden

⑤6 Für die Beurteilung der Patentfähigkeit  
in Betracht zu ziehende Druckschriften:

DE	26 32 277 A1
EP	01 24 615 A1
EP	00 44 192 A1
SU	16 81 297 A1

⑤4 Verfahren zur Erzeugung von NC-Steuerdaten

⑤7 Vorgeschlagen wird ein Verfahren zur Erzeugung von Steuerdaten aus einem in Form von jeweils mindestens eine Programmanweisung enthaltenden Programmsätzen abgefaßten NC-Programm für eine numerisch gesteuerte Maschine. Im voraus werden zunächst die NC-Funktionalitäten der im NC-Programm verwendeten Programmanweisungen, die zur Realisierung jeder NC-Funktionalität erforderlichen Einzelmaßnahmen sowie deren Ausführungsreihenfolge festgelegt. Die Zuordnungen von Programmanweisungen zu NC-Funktionalitäten werden in einer Syntaxtabelle, die Zuordnungen von Einzelmaßnahmen zu NC-Funktionalitäten in einer Funktionstabelle, die Ausführungsreihenfolge aller Einzelmaßnahmen in einer Jobtabelle abgelegt. Mit den Tabellen erfolgt die Steuerdatenerzeugung. Jeder Programmanweisung des NC-Programms wird entsprechend der Syntaxtabelle eine Funktionalität zugeordnet, dieser wiederum gemäß Funktionstabelle eine Einzelmaßnahme. Alle enthaltenen Einzelmaßnahmen werden abgespeichert und anschließend in ausführbare Maschinenanweisungen umgesetzt.



DE 43 21 631 A 1

**BEST AVAILABLE COPY**

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

BUNDESDRUCKEREI 11. 94 408 062/91

9/29

## Beschreibung

## Stand der Technik

Die Erfindung geht aus von einem Verfahren nach der Gattung des Hauptanspruchs. Verfahren dieser Art werden in nahezu allen gebräuchlichen NC-Steuerungen eingesetzt, um Bearbeitungsprogramme, die mehr oder weniger in Klartext gefaßt sind, in von der Maschine ausführbare Steuerdaten zu überführen. Üblicherweise bestehen die Bearbeitungsprogramme aus einer Folge von einzelnen NC-Sätzen, welche wiederum aus einzelnen NC-Worten aufgebaut sind. Zur Überführung in maschinenausführbare Daten dient in der Regel eine Satzaufbereitung oder eine vergleichbare Vorrichtung, welche das Bearbeitungsprogramm satzweise ausliest und wortweise in NC-Steuerdaten übersetzt. Der Zusammenhang zwischen den einzelnen NC-Worten und den zugehörigen von der Maschine auszuführenden Aktionen ist dabei im allgemeinen im Satzaufbereitungsprogramm festgelegt. Sollen bei einer solchen üblichen Steuerung einzelne NC-Worte verändert, erweitert oder neu definiert werden, ist der Einsatz eines hierfür qualifizierten Programmierers erforderlich, welcher die gewünschten Änderungen in Maschinensprache im Satzaufbereitungsprogramm vornimmt. Nachteilig an einer solchen Steuerung ist, daß vor allem dann, wenn das Satzaufbereitungsprogramm häufig geändert wurde, die Programmstruktur sehr komplex werden kann. Dadurch werden einerseits weitere Änderungen zunehmend schwieriger, zum anderen geht eine komplexe Programmstruktur auf Kosten der Bearbeitungsgeschwindigkeit.

Aus der EP 124 615 ist ein Verfahren zum flexiblen Erzeugen von NC-Programmen in beliebig vorgebbaren Formaten bekannt. Dabei werden von der Maschine auszuführende Bearbeitungsmuster zunächst durch Parameter beschrieben und anschließend mit Hilfe zweier Tabellen, von denen die erste die Parameter mit maschineninternen Funktionscodes, die zweite die Funktionscodes mit den NC-Programmsätzen in Verbindung setzt, in ein NC-Programm überführt. Durch vergleichsweise einfaches Ändern der Tabellen kann hier die Erzeugung des NC-Programms gut beeinflußt werden. So ist zum Beispiel sowohl die Hinzunahme weiterer Befehle, wie auch die Neudefinition beliebiger Anweisungen in einem anderen Format möglich.

Aufgabe der vorliegenden Erfindung ist es, ein Verfahren anzugeben, das gleichzeitig eine flexible Satzverarbeitung der einzelnen Sätze eines NC-Programmes liefert und schnell ist.

Die Aufgabe wird gelöst durch ein Verfahren mit den Merkmalen des Hauptanspruchs. Durch Verwendung von in Klartext programmierbaren Tabellen, welche die Zuordnung von NC-Programmsatz und von der Maschine auszuführender Anweisung schrittweise herstellen, ist die vorgeschlagene Satzverarbeitung frei konfigurierbar. So gestattet eine Syntaxtabelle, daß das erfindungsgemäße Verfahren hinsichtlich der bei der NC-Programmerstellung zu verwendenden Programmiersprache an jede beliebige NC-Syntax anpaßbar ist. Durch systematische Auflistung der möglichen NC-Funktionen in einer Funktions- und in einer Jobtabelle ist es ohne weiteres möglich, neue NC-Funktionen einzufügen, alte NC-Funktionen durch neue Funktionen zu ersetzen oder zum Beispiel die Reihenfolge der Abarbeitung zu ändern. Das erfindungsgemäße Verfahren zur Verarbeitung führt, vor allem bei kurzen NC-Sätzen, zu einer deutlichen Verringerung der Blockzykluszeit, da nur Zyklen der tatsächlich programmierten NC-Funktionen ausgeführt werden. Das vorgeschlagene Verfahren eignet sich deshalb gut für Hochgeschwindigkeitsbearbeitungen.

Ein Ausführungsbeispiel der Erfindung wird nachfolgend in der Beschreibung und anhand der Zeichnung näher erläutert. Es zeigen

Fig. 1 ein Strukturschaubild einer Einrichtung zur Durchführung des vorgeschlagenen Verfahrens, Fig. 2 ein Bearbeitungsbeispiel in geometrischer Darstellung, Fig. 3 ein Beispiel für eine Syntaxtabelle, Fig. 4 ein Beispiel für eine Funktionstabelle, Fig. 5 ein Beispiel für eine Jobtabelle, Fig. 6 gemäß zum Beispiel der Fig. 2 erzeugte Joblisten.

## Beschreibung

Ein Strukturschaubild einer Anordnung zur Durchführung des erfindungsgemäßen Verfahrens zeigt Fig. 1. In einem mit der Bezugszahl 10 bezeichneten Teileprogramm-Speicher befindet sich ein von einer NC-Maschine 15 auszuführendes Bearbeitungsprogramm. Es weist eine übliche Struktur mit mehreren Sätzen, die wiederum aus einzelnen Worten bestehen, auf und wurde in bekannter Weise erstellt, zum Beispiel mittels einer automatischen Programmerstellungsvorrichtung gemäß Patent EP-124 615. Durch einen Joblistenerzeuger 30 sowie einen Joblisteninterpretierer 40 wird das im Speicher 10 abgelegte Bearbeitungsprogramm in von der Maschine 15 ausführbare Steuerdaten übersetzt, die einem Interpolator 12 und von diesem der Maschine 15 zugeführt werden. Sowohl Joblistenerzeuger 30 als auch Joblisteninterpretierer 40 verfügen jeweils über eine eigene zentrale Prozessoreinheit CPU1 beziehungsweise CPU2 sowie über je einen eigenen Arbeitsspeicher. Über eine Datenverbindung 26 stehen Joblistenerzeuger 30 beziehungsweise Joblisteninterpretierer 40 miteinander sowie mit den übrigen, nachfolgend beschriebenen Funktionselementen in Verbindung. Im einzelnen kommuniziert der Joblistenerzeuger 30 über die Datenleitung 26 ferner mit dem Teileprogramm-RAM-Speicher 10, mit einem RAM-Speicher 11 zur Speicherung einer Syntax-, einer Funktions-, und einer Jobtabelle, mit einem RAM-Speicher 14 zur Speicherung einer Jobliste, in welcher blockweise Adressen von durch den Joblisteninterpretierer 40 abzuarbeitender Jobs abgespeichert sind, mit einem RAM-Speicher 19 zur Speicherung einer aus den im Speicher 11 abgelegten Tabellen erzeugten internen Struktur, sowie mit einem Bedienfeld, über welches eine Bedienperson Eingaben oder Veränderungen der in den Speichern und den Zentraleinheiten vorhandenen Daten vornehmen kann. Zur Interpolation der zur Übergabe an die Maschine 15 bestimmten Steuerdaten dient ein Interpolator 12, der hierzu eine eigene Prozessoreinheit CPU3 aufweist. In einem (Nur-Lese-)ROM-Speicher 13 befinden sich die Betriebsprogramme für Joblistenerzeuger 30, Joblisteninterpretierer 40 und Interpolator 12.

Die gewählte Aufteilung der in Fig. 1 dargestellten Struktur dient zum Zwecke einer einfacheren Erläuterung. Eine tatsächliche Realisierung kann von der dargestellten Struktur stark abweichen und wird von einem Fachmann ohne Mühe in Anpassung an die gegebene Hardwarestruktur vorgenommen. Beispielsweise ist es oft zweckmäßig, Joblistenerzeuger 30 und Joblisteninterpreter 40 innerhalb einer übergeordneten Steuerungsprozessorstruktur zu realisieren. Die in Fig. 1 auf drei Prozessoren CPU1, CPU2 und CPU3 verteilten Aufgaben können auch von einer einzigen CPU durchgeführt werden, beispielsweise durch Verwendung eines Multitasking-fähigen Betriebssystems.

Nachfolgend wird die Funktion der in Fig. 1 dargestellten Struktur anhand des in Fig. 2 gezeigten Bearbeitungsbeispiels erläutert. Ein mögliches zugehöriges, im Speicher 10 abgelegtes NC-Teilprogramm zeigt die nachfolgende Tabelle 1.

N10	G1	Z10	F100
N20	X10	Y30	
N30	X30	F200	
N40	G2	Y10	R10
N50	G1	X20	
N60	X10	Y30	
N70	M30		

Tabelle 1

Es weist sieben mit N10 bis N70 bezeichnete Programmsätze auf. Im ersten Programmsatz N10 wird mit dem Wort G1 festgelegt, daß die nachfolgende Bewegung eine lineare Bewegung ist, mit dem Wort Z10 wird die Koordinate auf der Zeichenebene senkrechten Z-Achse auf den Wert 10 festgelegt, mit dem Wort F100 wird die Vorschubgeschwindigkeit auf den Wert 100 eingestellt. Die nachfolgenden Programmzeilen N20 bis N50 definieren das Anfahren der Punkte P1 bis P4. Gemäß der zweiten Programmzeile N20 wird durch die Worte X10 und Y30 dem Punkt P1 die Koordinate  $X = 10$ ,  $Y = 30$  in einem X/Y-Koordinatensystem zugewiesen. Gemäß der dritten Programmzeile N30 wird dem Punkt P2 durch das Wort X30 die Koordinate  $X = 30$  zugewiesen, die Y-Koordinate bleibt unverändert. Ferner wird durch das Wort F200 dieser Bewegung die Vorschubgeschwindigkeit 200 zugeordnet. In der vierten Programmzeile N40 wird durch das Wort G2 festgelegt, daß die nachfolgende Bewegung eine Kreisbewegung ist, durch das Wort Y10 wird festgelegt, daß der Endpunkt die Y-Koordinate 10 hat, das Wort R10 definiert den Radius der Kreisbewegung auf den Wert 10. In der fünften Programmzeile N50 bewirkt das Wort G1 wiederum eine lineare Bewegung, das Wort X20 definiert die X-Koordinate des Endpunktes dieser Bewegung auf den Wert 20. Die sechste Programmzeile N60 definiert durch die Worte X10 und Y30 den Punkt P1 auf die Koordinaten  $X = 10$  und  $Y = 30$ . Die siebte Programmzeile N70 enthält mit dem Wort M30 einen Befehl, welcher das Programm beendet.

Das Teilprogramm ist nicht notwendig, wie in Fig. 2 gezeigt, in einer genormten Syntax abgefaßt. Es kann ebenso in einer vom Anwender mittels der Syntaxtabelle im Speicher 11 festgelegten, eigenen Sprache vorliegen. Die einzelnen Sätze des Programmes im Beispiel Fig. 2 N10 bis N70 werden vom Joblistenerzeuger 30 wortweise ausgelesen. Zu jedem Wort ermittelt der Joblistenerzeuger 30 in einem ersten Schritt mit Hilfe der im Speicher 11 abgelegten Syntaxtabelle zunächst die zugehörige NC-Funktionalität. Hierzu ist die Syntaxtabelle eine Zuordnungsliste, welche jedem in der Anwendersyntax möglichen NC-Wort eine für die Steuerung verständliche NC-Funktion zuordnet. Ein Beispiel für eine Syntaxtabelle zeigt Fig. 3. Sie enthält alle in der benutzten Syntax möglichen NC-Worte, in Fig. 3 als "Syntaxstring" bezeichnet, im Beispiel gemäß Fig. 2 beziehungsweise Tabelle 1 sind dies die Worte: X, Y, Z, R, F, N, F, M30, G1, G2, sowie den diesen Worten zugeordneten NC-Funktionalitäten. Dabei wird allgemein zwischen Funktionen, welche eine unmittelbare Maschinenaktion bewirken, wie Vorschub, ("Ncf\_Vorschub"), Programmende ("Ncf\_Progende"), Linearvorschub ("Ncf\_linear"), Kreisbewegung ("Ncf\_Zirk\_re"), als auch Parameterzuordnungen, die nicht eine sofortige Maschinenaktion zur Folge haben, wie Radius, ("P\_Radius"), Achskoordinate ("P\_Achse 1,2,3"), Vorschubgeschwindigkeit ("P\_Vorschub"), Satznummer ("P\_Satznr") unterschieden. Daneben enthält die Syntaxtabelle zweckmäßig noch weitere Angaben über den Zusammenhang zwischen NC-Wort und NC-Funktion, wie insbesondere eine Angabe über den NC-Worttyp — "Parameter" oder "Funktion" —, über den Geltungsbereich eines Parameters — "satzlokal", wenn ein Parameter nur der zugehörigen NC-Funktion bekannt ist, oder "satzglobal", wenn alle NC-Funktionen darauf Zugriff haben —, oder über die modale Gruppe einer NC-Funktionen — zum Beispiel "keine" oder "Gr\_Interpol", wenn eine Funktion zur Gruppe der Interpolationen gehört.

Im Teilprogramm-Beispiel Fig. 2 sind folgende NC-Funktionen vorhanden: F, G1, G2, M30, als Parameter: N, R, F, X, Y, Z. Der Joblistenerzeuger 30 ordnet nun gemäß Syntaxtabelle Fig. 3 zum Beispiel dem NC-Wort F die NC-Funktion "Ncf\_Vorschub", dem Parameter "R" die NC-Parameterfestlegung "P\_Radius" zu. Mit allen übrigen NC-Worten des Teilprogrammes der Fig. 2 wird analog verfahren.

In einem zweiten Schritt ordnet der Joblistenerzeuger 30 mit Hilfe der Funktionstabelle der gefundenen NC-Funktion die zur Durchführung dieser NC-Funktion erforderlichen Einzelmaßnahmen, diese werden nachfolgend als Job bezeichnet, zu. Die Funktionstabelle ist ebenfalls als Zuordnungsliste aufgebaut, ein Beispiel für eine Funktionstabelle zeigt Fig. 4. In der Spalte "Jobliste" sind die Jobs angeführt, welche zur Umsetzung der in der Spalte "NC-Funktion" aufgeführten NC-Funktion erforderlich sind. So erfordert die NC-Funktion "Ncf\_Vorschub" die Einzeljobs "set\_feedrate" und "crt\_feedrate", um die Vorschubgeschwindigkeit auf einem Bildschirm

anzuzeigen; die NC-Funktion "Ncf\_Progende", die Einzeljobs "finish preparation" um die Bearbeitung zu beenden sowie "finish-prog", um das Programm zu beenden; die NC-Funktion "Ncf\_Linear", die Einzeljobs "init\_linear\_block", um eine lineare Bewegung zu initialisieren, "prepare\_linear\_block", um eine lineare Bewegung vorzubereiten, "prepare\_linear\_kinematic", um die Kinematik einer Linearbewegung vorzubereiten, sowie "prepare\_linear\_profile", um ein rampenförmiges Bahngeschwindigkeitsprofil vorzubereiten; die NC-Funktion "Ncf\_circ\_re" die Einzeljobs "init\_circular\_block", um eine Kreisbewegung zu initialisieren, "check\_rog\_circle\_data", um die Kreisdaten zu überprüfen, "prepare\_circular\_block", um eine Kreisbewegung vorzubereiten, "prepare\_circular-kinematic", um die Kinematik für eine Kreisbewegung vorzubereiten, "prepare\_circular\_ramp\_profile", um eine rampenförmiges Bahngeschwindigkeitsprofil vorzubereiten. Neben der Zuordnung von Job zu NC-Funktion enthält auch die Funktionstabelle zweckmäßig noch weitere Angaben, zum Beispiel eine Angabe über die zu einer NC-Funktion erwarteten Parameter. Im Beispiel Fig. 4 ist es etwa bei der NC-Funktion "Ncf\_Vorschub" erforderlich, nachfolgend einen Parameter "P\_Vorschub", bei der NC-Funktion "Ncf\_Zirk\_re", nachfolgend einen Parameter "P\_Radius" für den Radius des Kreises, sowie die Parameter "P\_Interpol\_1/2" für eine Mittelpunktprogrammierung einzugeben.

In einem dritten Schritt wird mit Hilfe der Jobtabelle eine Abarbeitungsreihenfolge für die in den vorhergehenden Schritten gefundenen Jobs festgelegt. Dies kann zum Beispiel erfolgen, indem jedem Job in der Jobtabelle eine Ordnungszahl zugeordnet ist. Die Abarbeitung der zu einem NC-Block gehörenden Jobs erfolgt jeweils in der Reihenfolge der Ordnungszahlen der vorhandenen Jobs. Fig. 5 zeigt für das Programm-Beispiel gemäß Fig. 2 eine andere Jobtabelle, bei der die möglichen Jobs hierarchisch angeordnet sind. Ein in der Jobtabelle nach Fig. 5 weiter oben stehender Job wird stets vor einem weiter unten stehenden Job ausgeführt. So wird beispielsweise der Einzeljob "set\_feedrate", um eine Vorschubgeschwindigkeit zu setzen, stets, falls vorhanden, vor den Einzeljobs "prepare\_linear\_block", um eine lineare Bewegung vorzubereiten oder "prepare\_circular\_kinematic", um die Kinematik für die Kreisbewegung vorzubereiten, ausgeführt. Desweiteren ordnet die Jobtabelle jedem Job eine Adresse zu, unter welcher die zu einem Job gehörenden Maschinenanweisungen, welche in einem zweckmäßig ebenfalls im Speicher 11 angeordneten Jobverzeichnis abgespeichert sind, auffindbar sind.

Die so gefundenen Jobs werden in der im Arbeitsspeicher 14 angelegten Jobliste blockweise abgespeichert. Ein Block entspricht jeweils einem NC-Satz. Fig. 6 zeigt als Beispiel eine aus mehreren Blocks bestehende Jobliste, wie sie sich für das in Fig. 2 gezeigte Bearbeitungsbeispiel ergeben konnte. So enthält beispielsweise die dem ersten Programmsatz N10 entsprechende Jobliste als ersten Einzeljob den Befehl "set\_feedrate", um eine Vorschubgeschwindigkeit zu setzen, als zweiten Einzeljob den Befehl "init\_linear\_block", um einen linearen Block zu initialisieren, als dritten Einzeljob den Befehl "prepare\_linear\_block", um einen linearen Block vorzubereiten, als vierten Einzeljob den Befehl "prepare\_linear\_kinematic", um die Kinematik für eine Linearbewegung vorzubereiten, als fünften Einzeljob den Befehl "prepare\_linear\_ramp\_profile", um ein rampenförmiges Bahngeschwindigkeitsprofil vorzubereiten, sowie als sechsten Einzeljob den Befehl "crt\_feedrate", um die Vorschubgeschwindigkeit anzuzeigen. In gleicher Weise enthält der Arbeitsspeicher 14 zu den weiteren Sätzen N20 bis N70 des in Tabelle 1 wiedergegebenen Bearbeitungsprogrammes jeweils eine Liste mit den dem Programmsatz entsprechenden Einzeljobs. Das Programm wird entgegen der Pfeilrichtung abgearbeitet.

Die Abarbeitung der Jobliste erfolgt durch den Joblisteninterpreter 40. Er liest entsprechend der vorgegebenen Reihenfolge die in der Jobliste abgelegten Jobadressen aus und sucht die zugehörigen Maschinenanweisungen im Jobverzeichnis auf. Die gefundenen Anweisungen werden ausgeführt, die dabei ermittelten Daten an den Interpolator 12 übergeben, welcher sie interpoliert und der Maschine 15 zuführt.

Seine große Flexibilität bezieht das vorstehend beschriebene Verfahren insbesondere aus dem Gebrauch der im Speicher 11 abgelegten Tabellen. Diese gestatten in beliebiger Weise Veränderungen der Satzverarbeitung, Anpassungen an Anwender, welche eine eigene, nicht standardisierte Programmiersyntax verwenden, sind dadurch ebenso einfach wie zum Beispiel Anpassungen der vorhandenen Steuerung an eine neue Maschine.

Obgleich Änderungen der Satzverarbeitung leicht möglich sind, treten sie doch im Routine-Betrieb einer Steuerung, nachdem eine geeignete Syntax einmal festgelegt ist, vergleichsweise selten auf. Ständige Zugriffsmöglichkeit auf die im Speicher 11 abgelegten Tabellen muß deshalb nicht gegeben sein. Zur Beschleunigung des erfindungsgemäßen Verfahrens ist es in diesen Fällen zweckmäßig, einen Konfigurator vorzusehen, welcher bei Inbetriebnahme der Maschine 15 die im Speicher 11 abgelegten Tabellen in einem Konfigurationslauf miteinander verknüpft und die gewonnene interne Struktur in einem weiteren Speicher 19 ablegt. Während des Betriebes vereinfacht sich dadurch das Zuordnen der auszuführenden Jobs zu den einzelnen NC-Worten beziehungsweise NC-Sätzen indem erstere den letzteren in bekannter Weise direkt zugeordnet werden. Die Zwischenschritte des Auffindens des zur einer NC-Syntax gehörenden NC-Funktionen mittels der im Speicher 11 abgelegten Syntaxtabelle beziehungsweise des Auffindens der zu den NC-Funktionen gehörenden Jobs mittels der Funktionstabelle können entfallen. Da die im Speicher 19 abgelegte interne Struktur zudem nicht lesbar sein muß, und aus diesem Grunde zweckmäßig in Maschinensprache abgelegt ist, wird die Zeit zur Abarbeitung eines NC-Blockes erheblich verringert. Ein Generator zur Verknüpfung der Tabellen im Speicher 11 ist zweckmäßig innerhalb der zentralen Prozessoreinheit CPU1 des Joblistenerzeugers 30 realisiert.

#### Patentansprüche

1. Verfahren zur Erzeugung von Steuerdaten für eine NC-Maschine aus einem in Form von jeweils mindestens eine Programmanweisung enthaltenden Programmsätzen abgefaßten NC-Programm, gekennzeichnet durch folgende Schritte

- Festlegen der NC-Funktionalitäten der im NC-Programm verwendeten Programmanweisungen im voraus,
- Festlegen der zur Realisierung jeder NC-Funktionalität erforderlichen Einzelmaßnahmen im voraus

- Festlegen der Ausführungsreihenfolge aller Einzelmaßnahmen im voraus
  - Speichern der Zuordnungen von Programmanweisungen zu NC-Funktionalitäten in einer Syntaxta-  
belle
  - Speichern der Zuordnungen von Einzelmaßnahmen zu NC-Funktionalitäten in einer Funktionsta-  
belle
  - Speichern der Ausführungsreihenfolge aller Einzelmaßnahmen in einer Jobtabelle
  - Zuordnen wenigstens einer Funktionalität zu jeder Programmanweisung des NC-Programms ge-  
mäß Syntaxtabelle
  - Zuordnen wenigstens einer Einzelmaßnahme zu jeder Funktionalität gemäß Funktionstabelle
  - Abspeichern aller erhaltenen Einzelmaßnahmen
  - Umsetzen der abgespeicherten Einzelmaßnahmen in ausführbare Maschinenanweisungen.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß jede in der Jobtabelle abgelegte Einzelmaßnah-  
me eine Ordnungskennzeichnung aufweist, welche den Ausführungszeitpunkt dieser Einzelmaßnahme in  
bezug auf eine andere festlegt.
3. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die erhaltenen Einzelmaßnahmen in einer  
Jobliste in der Reihe ihrer Ordnungskennzeichnungen abgespeichert werden.
4. Verfahren nach Anspruch 2, dadurch gekennzeichnet, daß die erhaltenen Einzelmaßnahmen in der Reihe  
ihrer Ordnungskennzeichnungen abgearbeitet werden.
5. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, daß das Zuordnen einer  
Funktionalität zu jeder Programmanweisung und das Zuordnen von Einzelmaßnahmen zu jeder Funktionalität  
in einem Konfigurationslauf vor Abarbeitung eines NC-Programms durchgeführt wird, und daß die  
dabei ermittelte interne Struktur in einem Speicher 19 abgelegt wird.
6. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß jeder Einzelmaßnahme ausführbare Maschi-  
nenanweisungen zugeordnet werden, welche in einem Jobverzeichnis abgelegt sind.
7. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die erhaltenen Einzelmaßnahmen in einer  
Jobliste gespeichert werden.
8. Verfahren nach Anspruch 7, dadurch gekennzeichnet, daß die Einzelmaßnahmen blockweise abgespei-  
chert werden, wobei zu jedem NC-Satz des NC-Programms ein Block angelegt wird.
9. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß die Einzelmaßnahmen in der Jobtabelle in der  
Reihenfolge ihrer Abarbeitungspriorität abgelegt sind.

Hierzu 4 Seite(n) Zeichnungen

- Leerseite -

**THIS PAGE BLANK (USPTO)**

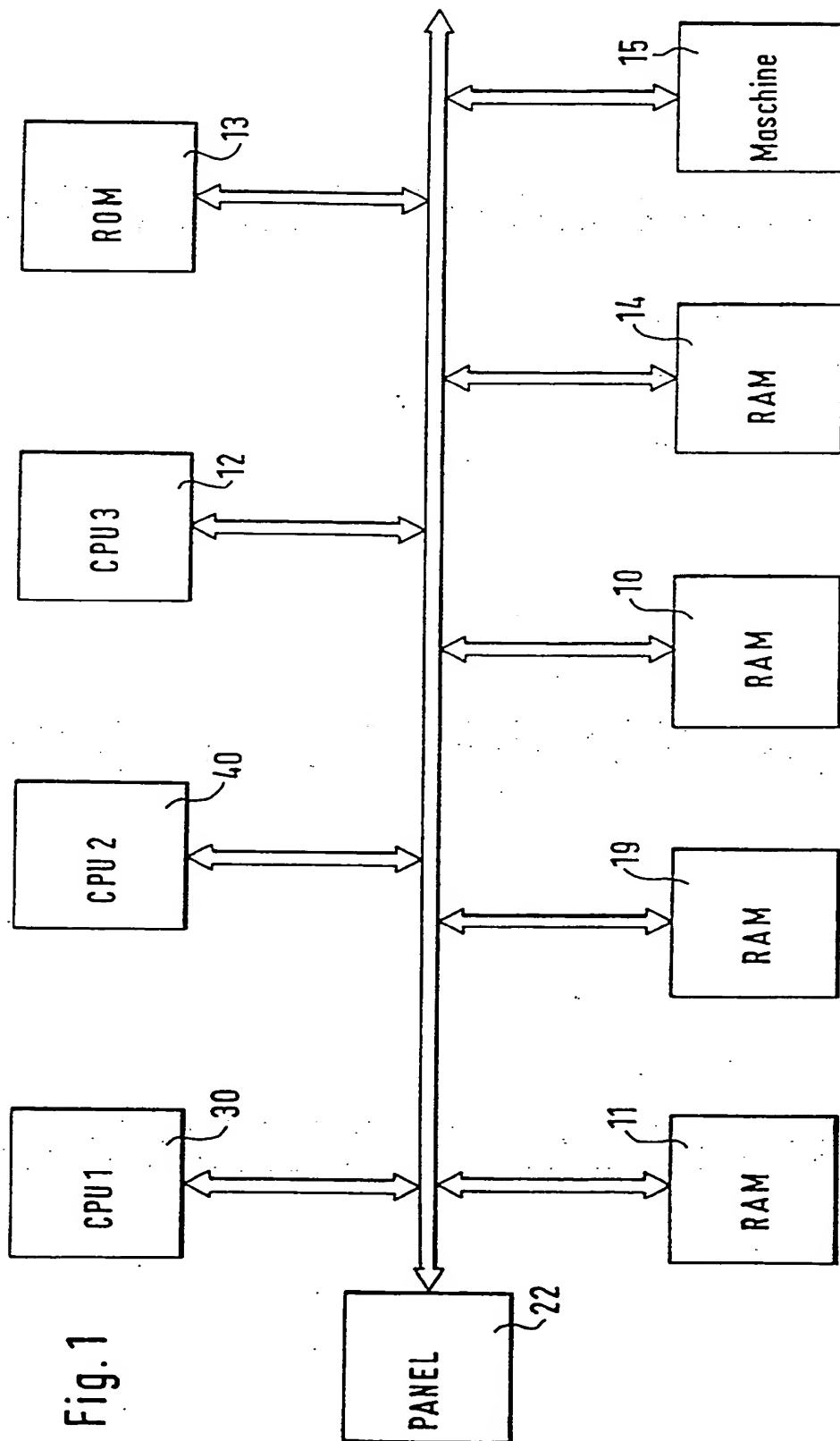


Fig. 1

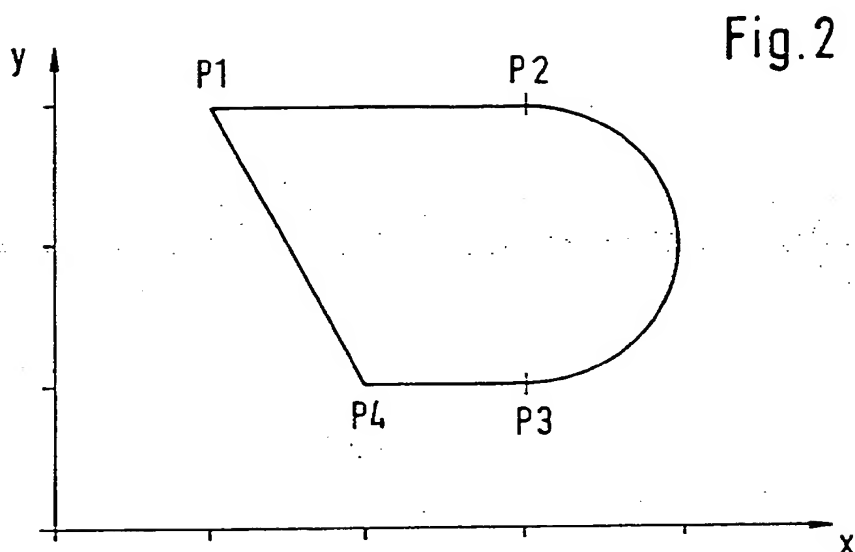


Fig.3

Syntax-string	Nc Wort-typ	Funktions-oder Para-metername	Parameter-typ	Geltungs-bereich	Modale Gruppe
X	Parameter	P- Achse-1	double	satzglobal	
Y	Parameter	P- Achse-2	double	satzglobal	
Z	Parameter	P- Achse-3	double	satzglobal	
R	Parameter	P- Radius	double	satzlokal	
F	Parameter	P- Vorschub	double	satzglobal	
N	Parameter	P- Satznr	int	satzglobal	
F	Funktion	Ncf-Vorschub			keine
M30	Funktion	Ncf- Progende			keine
G1	Funktion	Ncf-Linear			Gr-Interpol
G2	Funktion	Ncf-Zirk-re			Gr-Interpol

Nc-Funktion	Jobliste	Erwartete Parameter
Ncf_Vorschub	set_feedrate, crt_feedrate	P_Vorschub
Ncf_Progende	finish_preparation finish_prog	
Ncf_Linear	init_linear_block, prepare_linear_block, prepare_linear_kinematic, prepare_linear_ramp_profile	
Ncf_Zirk_re	init_circular_block, check_prog_circle_data, prepare_circular_block, prepare_circular_kinematic, prepare_circular_ramp_profile	P_Radius P_Interpol_1,

Fig. 4

```

...
finish_preparation
set_feedrate
init_linear_block
init_circular_block
...
check_prog_circle_data
prepare_linear_block
prepare_circular_block
prepare_linear_kinematic
prepare_circular_kinematic
prepare_linear_ramp_profile
prepare_circular_ramp_profile
...
crt_feedrate
finish_prog
...

```

Fig. 5

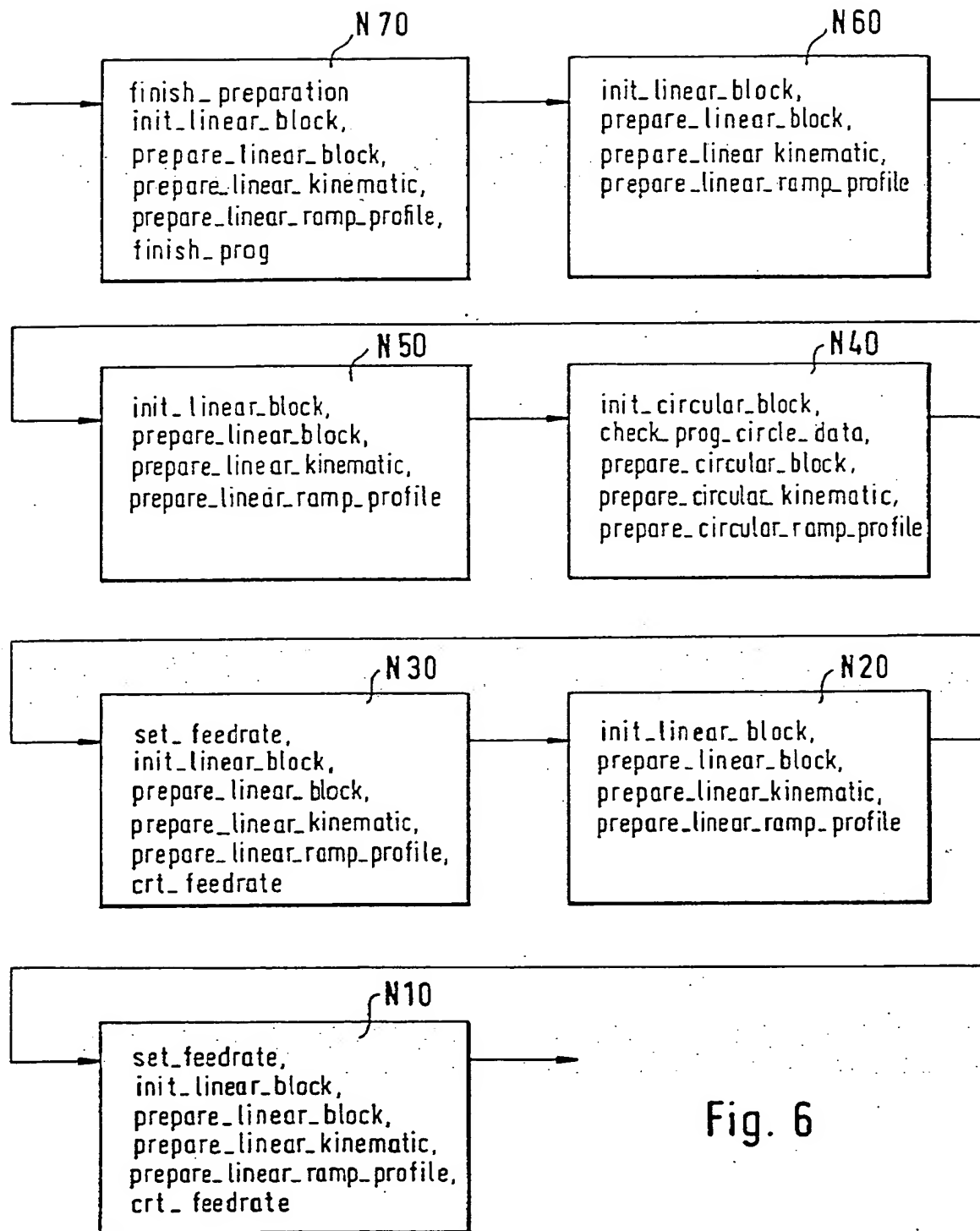


Fig. 6